Un environnement de programmation visuelle utilisant la métaphore du théâtre

Isabelle Borne

UN ENVIRONNEMENT DE PROGRAMMATION VISUELLE UTILISANT LA MÉTAPHORE DU THÉATRE

Isabelle BORNE

Résumé : cette étude se place dans le contexte de l'initiation à la programmation par objets. Notre objectif tente de répondre à la question : « Comment donner à des non-informaticiens la possibilité de programmer des applications graphiques interactives à des fins pédagogiques ? ». Une première expérience d'utilisation de la programmation par objets avec des instituteurs (Borne & Girardot 91) nous a permis de mettre en évidence l'importance de l'interaction avec les objets et les difficultés d'ordre syntaxique et conceptuel pour la création de nouvelles classes d'objets. Maintenant, elle nous amène à proposer aux élèves des collèges et aux éducateurs un environnement de programmation visuelle.

Une métaphore a été ajoutée au paradigme objet, celle de la mise en scène de théâtre pour, d'une part, aider au découpage d'une application et d'autre part augmenter l'interaction avec les objets. De plus, la programmation proprement dite concernant les répliques des acteurs s'effectue avec une syntaxe visuelle.

1. INTRODUCTION

L'introduction de l'informatique dans les collèges peut prendre des formes et des buts très divers. Les environnements de programmation constituent une approche assez large permettant non seulement aux éducateurs de réaliser des applications pédagogiques, mais également aux élèves d'aborder la programmation et ses concepts informatiques. La programmation par objets a fait ses preuves non seulement sur le marché industriel du génie logiciel (Meyer 90) mais également bien avant en milieu scolaire (Aubé & Bracke 87).

Cependant, l'enseignement de la programmation par objets à des non-informaticiens pose le double problème de la familiarisation avec un langage de programmation, voire avec un environnement de programmation, et celui du guide méthodologique pour la conception et le découpage des applications. Une première expérience d'utilisation de la programmation par objets (avec Smalltalk) avec des instituteurs (Borne & Girardot 91) nous a permis de mettre en évidence l'importance de l'interaction avec les objets et les difficultés d'ordre syntaxique et conceptuel rencontrées au moment de la création de nouvelles classes d'objets. Le discours d'une correspondance naturelle entre les concepts ou les objets du monde réel et ceux du système n'est valable que pour des problèmes très simples et ne permet pas de rendre explicite tous les liens entre les objets. La difficulté du

paradigme objet se situe souvent dans l'extraction des concepts du problème à résoudre, dans leurs relations et dans leurs fonctionnalités. La question se pose donc de plonger le paradigme objet dans le cadre plus large d'une métaphore qui apporte un début de solution au problème de décomposition d'une application.

Notre étude vise le public des collèges et propose un environnement de programmation par objet visuel s'appuyant sur la métaphore du théâtre. Cette métaphore proposée initialement dans le système « Programming by Rehearsal » (Gould & Finzer 84) est reprise ici et généralisée à tous les niveaux de décomposition d'un problème. La métaphore du théâtre a pour but de fournir un cadre plus rigide que celui de l'environnement Smalltalk de base, pour les tâches de décomposition d'une application et d'organisation des objets. Notre environnement se présente comme une couche graphique au dessus de Smalltalk-80. Il pallie au problème de la connaissance du langage Smalltalk - qui subsistait dans le système de Gould et Finzer pour développer des applications assez complexes - en offrant un langage de programmation visuelle pour la définition et la représentation des envois de message.

Après avoir présenté la métaphore du théâtre, nous décrivons l'architecture générale de notre système et son fonctionnement, puis nous terminons par un aperçu du langage de programmation visuelle.

2. LA MÉTAPHORE DU THÉÂTRE

La métaphore de la mise en scène d'une pièce de théâtre, pour modéliser la décomposition d'une application, ne va pas résoudre les difficultés d'extraction des concepts du problème et d'identification de leurs relations et de leurs fonctionnalités, mais va permettre de rendre familiers certains concepts non-familiers de la programmation et de guider leur mise en oeuvre dans un cadre conceptuel et fonctionnel.

Les éléments de la métaphore du théâtre sont les suivants :

- la *pièce* représente une application réalisant un problème donné ;
- le *scénario* représente le découpage d'une pièce en scènes. Il correspond à une certaine mise en scène, c'est-à-dire au point de vue d'un metteur en scène pour la réalisation de la pièce. Par conséquent, plusieurs scénarios peuvent être attachés à une même pièce, représentant autant de solutions possibles pour le problème principal ;
- la *scène* est l'élément de découpage de base. Elle représente une étape dans la réalisation de la pièce. Nous pouvons la comparer à la notion informatique de module, car les scènes sont conçues indépendamment les unes des autres. Leur enchaînement au sein d'un scénario figurera le déroulement de l'application. De plus, à chaque scène est associé un *script* qui est l'ensemble des répliques à jouer dans la scène ;
- le *décor* est un objet graphique passif attaché à une scène. Par exemple, un fond d'écran est un décor ;

- l'*acteur* est un élément actif de la pièce, il correspond à une instance de classe dans le paradigme objet. L'audition d'un acteur consiste à lui faire jouer une réplique, c'est l'envoi d'un message à un objet ;
- la *réplique* est une action réalisable par un acteur, elle correspond à une méthode Smalltalk ;
- le *costume* est un objet graphique attachable à un acteur. C'est la représentation visuelle d'un acteur à un moment donné.

Par ailleurs, les notions de classe et de catégorie du système Smalltalk bien que présentes, n'interviennent pas directement dans la métaphore. La création d'un acteur s'effectue par copie d'un autre acteur et un éditeur spécifique permet de faire l'ajustement de ses composantes. Par ailleurs, les acteurs sont regroupés dans des *troupes* qui constituent les bibliothèques des objets disponibles.

3. DESCRIPTION GÉNÉRALE DU SYSTÈME

L'environnement du Théâtre tourne sous Smalltalk-80 (version 4) sur station de travail avec X-Window, sur micro-ordinateur compatible avec Windows ou sur Apple Macintosh.

3.1 Uniformité de présentation et d'interaction avec les objets

La présentation des objets du système est uniforme (voir Figures 1 et 2). Elle s'effectue dans une fenêtre dont la partie gauche est réservée aux boutons des choix possibles et dont la partie droite est utilisée pour montrer sous forme d'icône les objets correspondant au bouton choisi. Le choix de l'icône a pour effet d'ouvrir une fenêtre de présentation pour cet objet.

Par exemple, la sélection dans la figure 2 de l'icône nommée « Initialisation » provoquera l'ouverture d'une fenêtre de présentation de la scène « Initialisation » analogue à la fenêtre de présentation du scénario. De même, la création d'un nouvel objet consiste simplement à sélectionner l'icône prévue à cet effet, par exemple l'icône « NouvelleScène » pour ajouter une scène au scénario 1 de la pièce Animation. Les choix du mode de présentation des objets et du mode d'interaction avec les objets doivent respecter certaines règles d'uniformité afin que la compréhension du système et sa manipulation soient suffisamment simples pour le non-informaticien.

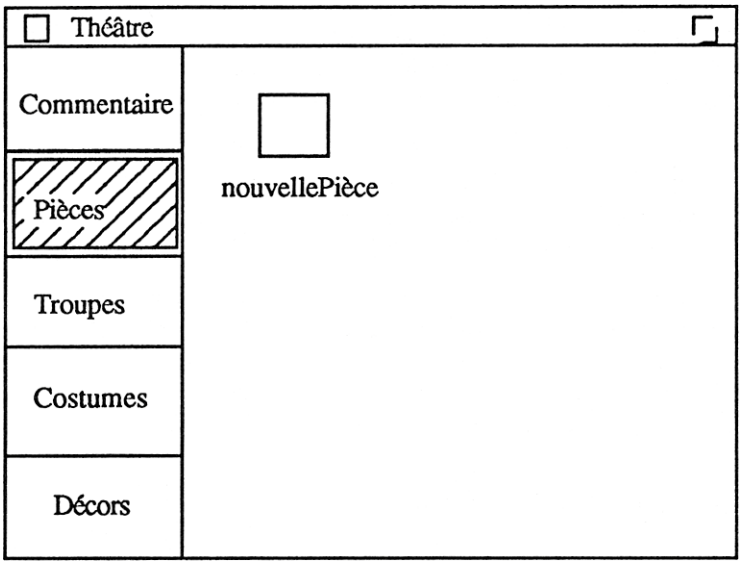

Figure 1. La fenêtre du Théâtre

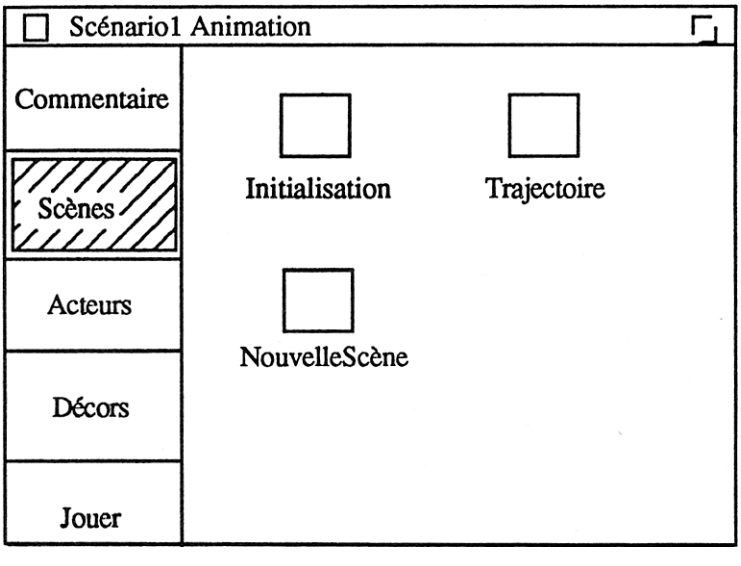

Figure 2. La fenêtre du premier scénario de la pièce Animation

3.2 Visibilité et documentation

Notre attention s'est surtout portée sur deux points : d'une part, la visibilité des objets manipulés et leur interaction et, d'autre part, sur la documentation ou les commentaires associés aux objets.

Tous les objets utilisés (Théâtre, Pièce, Scénario...) sont visibles et interactifs, l'utilisateur peut à tout moment désigner l'objet qui l'intéresse à l'aide de la souris et par exemple :

- obtenir un commentaire ou de l'aide concernant l'objet ;
- obtenir les objets du niveau suivant (par exemple obtenir toutes les scènes d'un scénario comme le montre la figure 2) ;
- observer ou modifier la structure de l'objet ;
- tester le comportement d'un objet, lui ajouter des éléments (scénario, scène) ou ajouter une réplique à un acteur.

De plus, l'utilisateur peut et doit valider chaque étape de son développement. Un bouton « Jouer » est disponible au niveau du scénario, de la scène et de la réplique. Il permet d'observer le comportement d'un acteur pour une réplique, d'observer l'enchaînement des répliques contenues dans le script d'une scène ou l'enchaînement des différentes scènes.

Nous espérons également que l'utilisateur prenne l'habitude de commenter ses programmes. Pour cela nous mettons à sa disposition des commentaires structurés. En effet, nous avons des informations de natures diverses concernant un acteur qu'il est bon de dissocier, par exemple :

- information générale concernant le type de l'acteur ;
- information spécifique sur son rôle de l'objet dans une scène ;
- information sur sa structure et sur ses répliques.

4. LA PROGRAMMATION VISUELLE

La programmation visuelle consiste à spécifier graphiquement les relations entre les objets d'une application grâce à un éditeur graphique plutôt qu'un éditeur de texte (Shu 88). Elle tente de libérer le programmeur de certaines tâches d'écriture de code pour lesquelles le système peut facilement le remplacer. Dans le paradigme objet, ce type de programmation est particulièrement bien adapté puisque des objets interactifs réagissant à des envois de messages y sont manipulés (Borne & Pachet 92).

De même qu'en Smalltalk la définition du corps d'une méthode consiste à combiner des envois de messages à des objets, ici, la définition du corps d'une réplique consiste à combiner des auditions d'acteurs sur des répliques. Les répliques possèdent comme les autres objets de l'environnement une fenêtre de présentation standard, ainsi qu'une représentation visuelle iconique étiquetée avec le nom de l'acteur et celui de la réplique. Sur la figure 3 nous voyons que la réplique « vaEtVient » est composée de trois autres répliques : trois auditions de l'acteur MaBoule et une audition de l'acteur Visu.

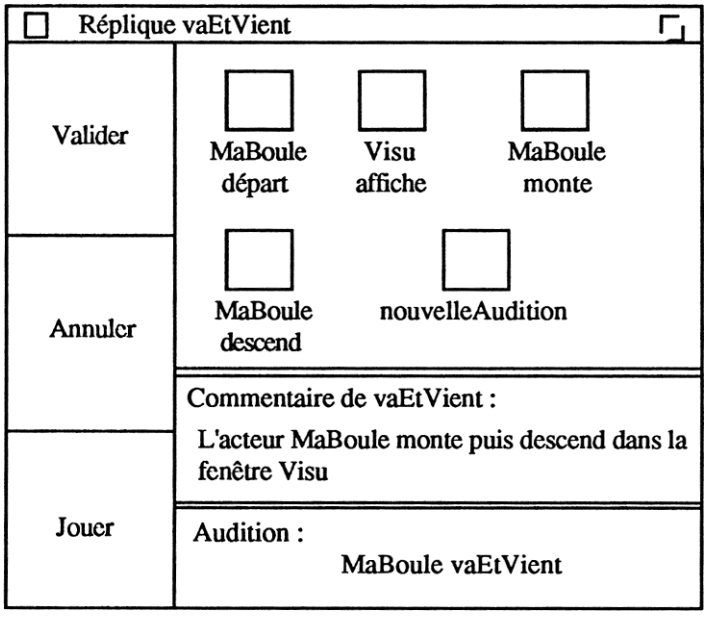

Figure 3. Fenêtre de présentation de la réplique vaEtVient

L'ajout d'une audition au corps d'une autre réplique s'effectue en sélectionnant l'icône « nouvelleAudition ». L'audition à ajouter est ensuite construite en désignant l'acteur concerné (dans une troupe ou dans une scène) et en choisissant une réplique (dans la fenêtre de présentation de l'acteur) parmi celles qu'il a déjà apprises.

Lors de la programmation visuelle d'une réplique, l'environnement théâtral engendre automatiquement le code Smalltalk correspondant, qui est transparent à l'utilisateur. Pour tous les types de données de base (caractères, chaînes, nombres) nous conservons les classes Smalltalk correspondantes qui sont regroupées dans une troupe nommée TroupeDeBase.

Dans son état actuel, notre programmation visuelle ne donne pas la possibilité d'utiliser des structures de contrôle. Il est possible de le faire en modifiant directement le code Smalltalk produit, mais ce n'est justement pas le but recherché. L'enchaînement des répliques est donc séquentiel pour l'instant et il est visualisé par la juxtaposition des icônes. Avant d'aller plus loin dans cette étape, nous avons besoin de valider le développement déjà effectué d'une part et, d'autre part, nous étudions un langage de programmation visuelle pour Smalltalk directement.

5. CONCLUSION

Notre environnement de programmation n'a pas bien sûr le pouvoir d'expression du système Smalltalk. Cependant, il permet de maîtriser les définitions des concepts et de leurs savoir-faire et d'apprendre à répartir et à combiner les activations de ces savoir-faire dans des modules (les scènes). Il doit être considéré

comme le petit bassin dans lequel on apprend à nager en sécurité avant de se lancer dans le grand bassin.

Actuellement l'architecture générale de notre environnement est réalisée, c'est-à-dire, d'une part, la mise en place des concepts principaux de la métaphore de mise en scène théâtrale qui assurent le découpage conceptuel et, d'autre part, l'interface visuelle, composée des fenêtres de présentation des objets et du langage visuel de base pour définir les répliques, qui assure l'interaction avec les concepts.

La métaphore du théâtre est bien adaptée à la conception et au développement de séquences d'animation, car ce type de sujet se prête aisément au déroulement d'un scénario. Il nous faut maintenant, en plus de la poursuite du développement du système, tester cette métaphore sur d'autres types d'applications afin d'en cerner les limites. Par ailleurs, le choix d'une représentation visuelle interactive facilite l'interaction avec les objets dans le cadre d'une programmation par objets, mais il n'exclue pas le problème du passage de cette représentation à une représentation textuelle correspondante « lisible » et vice versa .

Isabelle BORNE
E.H.E.I.
Université René Descartes
45, rue des Saints-Pères
75006 Paris - FRANCE

Téléphone : **(+ 33 1) 47 03 31 27**
Fax : **(+33 1) 42 86 22 31**
E-mail : **ib@ehei.ehei.fr**

RÉFÉRENCES

M. AUBÉ & D. BRACKE (1987) « Apprendre à utiliser l'environnement Smalltalk-80 au secondaire », Journal canadien *BIP-BIP*, n° 47, décembre.

I. BORNE & C. GIRARDOT (1991) « Object-oriented programming in the primary classroom », *Computers & Education*, vol. 16, n°1, p. 93-98.

I. BORNE & F. PACHET (1992) « From object-oriented design to visual programming », *European Journal of Engineering Education*, Carfax publishing company, vol.17 n° 2, p. 195-201.

L. GOULD & W.FINZER (1984) « Programming by Rehearsal », *Byte*, vol.9, n°6, June 1984, p. 187-210.

B. MEYER (1990) *Conception et programmation par objets*, InterEditions.

B.A. MYERS (1990) « Taxonomies of Visual Programming and Program Visualization », *Journal of Visual Languages and Computing*, vol.1, n°1, March 1990, p. 97-123.

N.C. SHU (1988) *Visual Programming*, Van Nostrand Reinhold editor.